# Jordanian University of Science and Technology

## Bee Conferences Management

**Final Project 1 Report Submitted to**
**The Department of Computer Science**
**Faculty of Computer and Information Technology**
**Jordan University of Science and Technology**

**In Partial Fulfillment of the Requirements for the Degree of**
**Bachelor of Science in**
**Computer Science**

Moayad Shloul 140072

Supervisor:
Dr. Muneer Bani Yassein
Mr. Mustafa Radaideh

# ACKNOWLEDGMENTS

I am writing to express my sincere appreciation for the invaluable support and guidance you provided during the completion of my graduation project. This experience has been truly rewarding, and your effective and professional contributions have played a crucial role in its success.

Your wealth of experience and guidance significantly contributed to the development of my scientific and technical skills. Always available to answer queries and provide direction, your support was instrumental.

I wish to convey deep gratitude for the efforts invested, **Eng. Mustafa Radaideh**.

I hope our collaboration will continue in the future, affording me further opportunities to benefit from your wealth of expertise.

Thank you once again for your support.

# Table Of Contents


**Introduction** ---- 5

1. **Overview:** ---- 5
2. **Abstract** ---- 6
3. **Objectives:** ---- 7

**Literature Review** ---- 8

1. **EasyChair**: ---- 8
2. **OpenConf:** ---- 9

**Analysis phase** ---- 10

1. **Requirements:** ---- 10
   - 1.1. **List of functional requirements (Agile methodology - Adjustable)** ---- 10
   - 1.2. **List of non-functional requirements** ---- 11
2. **How it's work:** ---- 12
   - 2.1. **SuperChair:** ---- 12
   - 2.2. **Chair:** ---- 12
   - 2.3. **Reviewer:** ---- 12
   - 2.4. **Author:** ---- 13
3. **Diagrams:** ---- 14
   - 3.1. **Use Case:** ---- 14
   - 3.2. **Class:** ---- 22
   - 3.3. **Activity:** ---- 24
   - 3.4. **Sequence:** ---- 26

**Design phase** ---- 32

1. **Entity Relationship:** ---- 32
   - 1.1. **Entity Relationship Diagram:** ---- 32
2. **Database Design:** ---- 34
   - 2.1. **Database schema:** ---- 34

**GUI** ---- 36

1. **Screenshots:** ---- 36
   - 1.1. **Home page:** ---- 36
   - 1.2. **Find CFPs page:** ---- 37
   - 1.3. **Conference CFP:** ---- 38

**1.4. Create CFP page:** ---- 39

**1.5. Create Conference page:** ---- 40

**1.6. Select Role page:** ---- 41

**1.7. Submit Paper page:** ---- 42

**References** ---- 44

**1. Links:** ---- 44

Appendices ---- 45

**1. Appendix A Code:** ---- 45

**1.1. Attachments Code:** ---- 45

**2. Appendix B Schedule:** ---- 46

**2.1. Table:** ---- 46

# Table Of Figures

Figure 1 EasyChair Home page ..... 8

Figure 2 OpenConf Home Page ..... 9

Figure 3 Use Case Diagram ..... 14

Figure 4 Class Diagram ..... 22

Figure 5 Activity Diagram ..... 24

Figure 6 SuperChair Sequence Diagram ..... 26

Figure 7 Chair sequence Diagram ..... 27

Figure 8 Reviewer Sequence Diagram ..... 28

Figure 9 Author Sequence Diagram ..... 29

Figure 10 Proceeding Manager Sequence Diagram ..... 30

Figure 11 Entity Relationship Diagram ..... 32

Figure 12 Database Schema ..... 34

Figure 13 home screen 1 ..... 36

Figure 14 home screen 2 ..... 36

Figure 15 CFP page ..... 37

Figure 16 conference CFP page ..... 38

Figure 17 Create CFP page. ..... 39

Figure 18 Create Conference Page ..... 40

Figure 19 Select Role page. ..... 41

Figure 20 Submit Paper page. ..... 42

Figure 21 Sign In page ..... 43

Figure 22 Sign Up page ..... 43

# Introduction

## 1. Overview:

In an era characterized by rapid technological advancements, the need for efficient and streamlined platforms for academic and professional collaboration has never been more crucial. Recognizing this demand, our graduation project aims to contribute to the facilitation of scholarly exchange and conference management through the development of a comprehensive and user-friendly website.
Our project draws inspiration from successful platforms, understanding the pivotal role they play in orchestrating seamless conferences and fostering intellectual discourse. The primary objective of our conference management website is to provide organizers, authors, and participants with a robust and intuitive toolset to manage every aspect of the conference lifecycle.
our graduation project endeavours to develop a cutting-edge Conference Management Website. Drawing inspiration from successful platforms like EasyChair.org, our project aims to offer a comprehensive and user-friendly interface for conference organizers, authors, reviewers, and participants alike.

## 2. Abstract

In an era defined by the relentless pace of innovation, the demand for seamless and technologically advanced solutions in academic and professional conferences has reached unprecedented heights. This project endeavours to address this need through the development of a state-of-the-art Conference Management Website.

The proposed conference management website boasts a range of features designed to enhance every aspect of the conference lifecycle. From a streamlined abstract submission and peer review process to a highly customizable conference configuration interface, the platform caters to the diverse needs of conference organizers, authors, reviewers, and participants. Key components include intuitive participant registration, real-time communication channels fostering networking and collaboration, and a robust system for managing keynote speakers.

Emphasis is placed on user-friendliness, with a responsive interface ensuring accessibility across various devices. Security and privacy are paramount, with stringent measures in place to safeguard sensitive information. The architecture is designed for scalability and optimal performance, ensuring a seamless user experience even during periods of high traffic.

Documentation and user support channels are integral components, providing administrators, organizers, and participants with the resources needed to navigate the platform effortlessly. The commitment to continuous improvement is woven into the project's fabric, with user feedback mechanisms driving iterative enhancements to the platform's functionality.

By undertaking this project, we aspire to contribute to the evolution of conference management, providing a one-stop solution that not only meets the immediate needs of organizers and participants but also sets the stage for adaptability and ongoing improvement. Our Conference Management Website aims to be a catalyst for efficient collaboration, knowledge exchange, and networking in the dynamic landscape of academic and professional conferences.

## 3. Objectives:

- **Abstract Submission and Review:**
    - Streamlined system for authors to submit abstracts.
    - Efficient peer review process for submitted abstracts.
- **Conference Configuration:**
    - Customizable setup for defining tracks, sessions, and event details.
    - Tools for organizers to tailor conferences to specific themes and requirements.
- **Participant Registration:**
    - Intuitive registration processes for conference attendees.
    - Personalized participant dashboards for easy event navigation.
- **User Roles and Permissions:**
    - Distinct user roles with customized permissions for administrators, organizers, reviewers, authors, and participants.
- **Robust Review System:**
    - Transparent and efficient peer review system.
    - Timely feedback mechanisms for authors and reviewers.
- **User-Friendly Interface:**
    - Responsive and accessible design for ease of use across devices.
    - Intuitive navigation to enhance user experience.
- **Security and Privacy Measures:**
    - Robust data security measures to safeguard user information.
    - Privacy features to protect sensitive data.
- **Scalability and Performance Optimization:**
    - Architecture designed for scalability to accommodate growing conferences and user bases.
    - Performance optimization for a seamless user experience.

By offering these services, Bee conference management website aims to provide a comprehensive solution for conference organizers and participants, ensuring a smooth and efficient experience throughout the entire conference lifecycle.

## 1. EasyChair:

what is EasyChair?

**EasyChair** is a web-based conference management software system. It has been used since 2002 in the scientific community for tasks such as organizing research paper submission and review. In 2012, EasyChair began offering an open access online publication service for conference proceedings. []

What service dose it provides?

- Conference management, including abstract submission, reviewing, program generation and many other components.
- VCS for organizing virtual and hybrid conferences.
- Conference registration, with or without online payment.
- Smart CFP for publishing calls for papers.
- Smart Slide for publishing conference presentations.
- Publishing conference proceedings and other volumes.
- Preprints.[]

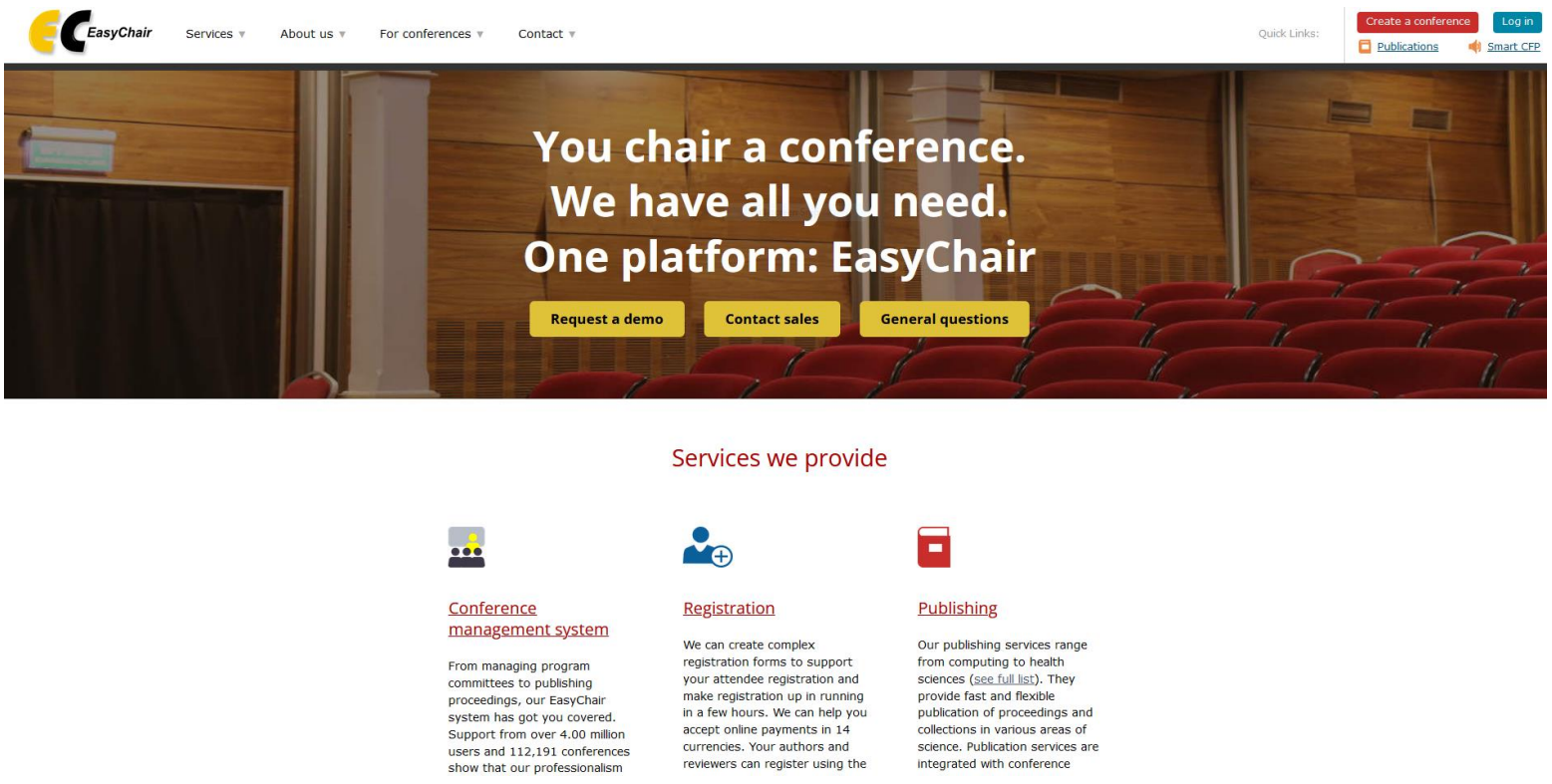


*Figure 1 EasyChair Home page*

## 2. OpenConf:

what is OpenConf?

**OpenConf** is an abstract management and peer-review web application used extensively by conferences and journals across a wide set of industries. The software is available both for download and cloud-based, with translations in over a dozen languages for author and reviewer interfaces.

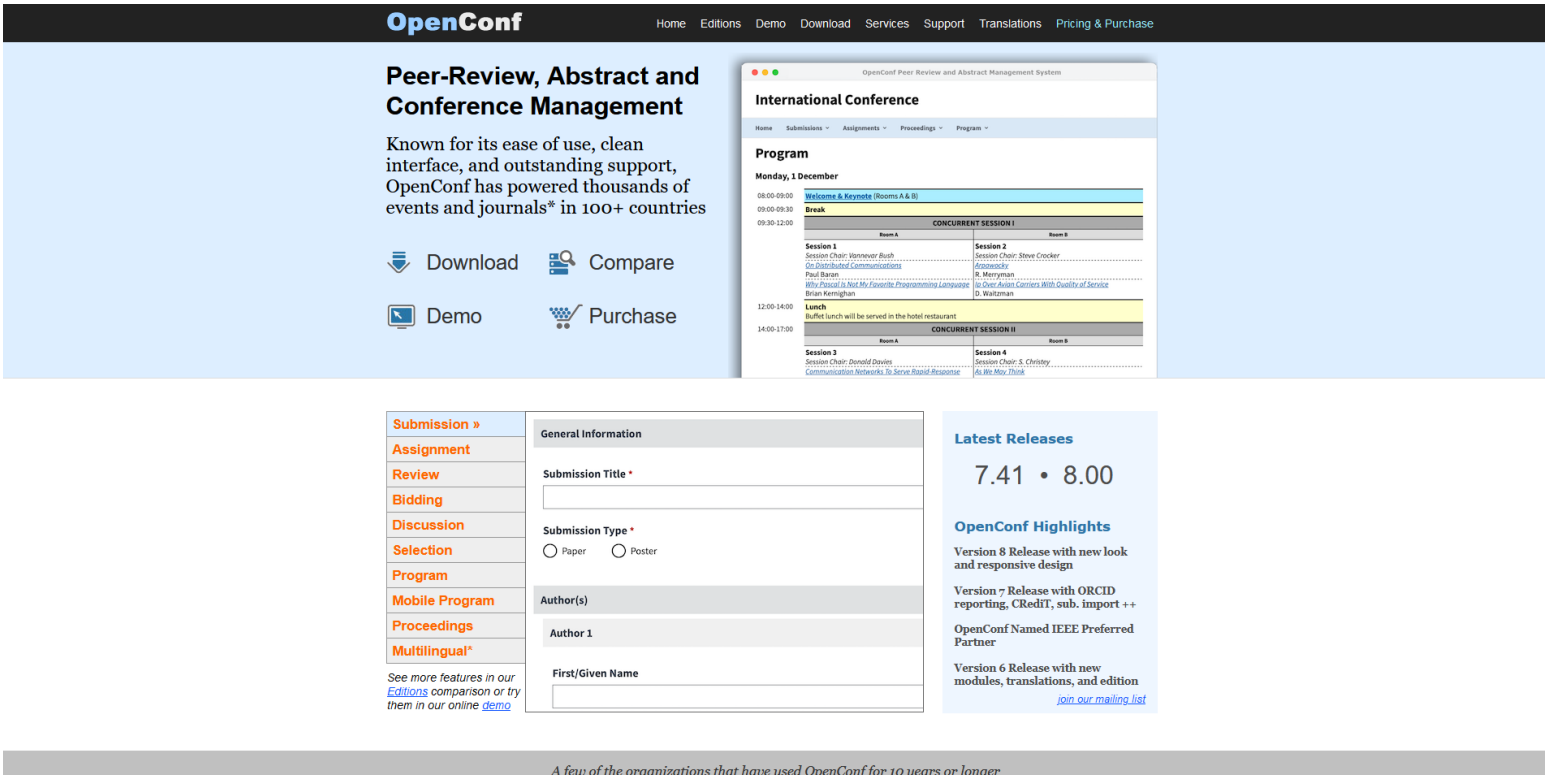


*Figure 2 OpenConf Home Page*

# Analysis phase

## 1. Requirements:

### 1.1. List of functional requirements (Agile methodology - Adjustable)

#### 1.1.1. Technical Admin:

Create, Delete, Block users.
Time scheduling.
Monitor database.
Monitor user activity.
Keep system up to date.
Fix system bugs.

#### 1.1.2. SuperChair

Create new conference.
Set tracks.
Determine the date of the conference.
Name conference.
Assign chair to tracks.

#### 1.1.3. Chair:

invite reviewers.
View reviewers.

#### 1.1.4. Reviewer:

View papers.
Read papers.
Give score.
Give feedback.
Give a hint (Optional).

#### 1.1.5. Author:

Upload paper.
Modify paper.
Submit paper.
Show the result of his paper.

## 1.2. List of non-functional requirements

### 1.2.1. Usability:

The user interface shall be intuitive, user-friendly, and accessible, catering to users of all levels of technical expertise.

### 1.2.2. Security:

User passwords shall be securely stored using strong encryption algorithms.

Access to sensitive data, such as user profiles.

### 1.2.3. Compatibility:

The website shall be compatible with major web browsers, including Chrome, Firefox, Safari, and Edge.

The system shall support the latest versions of these browsers to ensure optimal performance.

### 1.2.4. Data Privacy:

Personal user data, such as email addresses, shall be handled in compliance with data protection regulations (e.g., GDPR).

# 2. How it’s work:

## 2.1. SuperChair:

In the beginning, the **SuperChair** will goes through conference registration page to fill all conference details that’s represented by:

- Conference Name
- Conference Topic
- Conference description
- Conference date
- Conference time
- Conference Poster (If exist)
- Conference tracks

After that, his request will go to reviewing process to take approve by technical team.

If request approved **SuperChair** needs to take an action which choose **chairs** to tracks.

## 2.2. Chair:

Chair is an actor selected by **SuperChair**, he’s responsible for choosing reviewer to paper that submitted by **authors.**

Choosing reviewer follow 3 steps:

- Find reviewer specialized with specific track.
- Send invitation link.
- Accept invitation by reviewer.

## 2.3. Reviewer:

After reviewer accept invitation, he has a new responsibility that’s follow:

- Read author paper.
- Give it score.
- Give it feedback.
- Write reviewer confidentially when he gives score.

After the reviewer provides feedback, all data will be saved in database to calculate the total score from all reviewers to show to author when the review deadline is reached.

## 2.4. Author:

Author is the actor who has a paper wants to participate in a research paper within the conference, to do that's needs to:

- Register in a conference.
- Select track.
- Submit paper.

Then submission will go through the review process.

# 3. Diagrams:

## 3.1. Use Case:

### 3.1.1. Use Case Diagram:

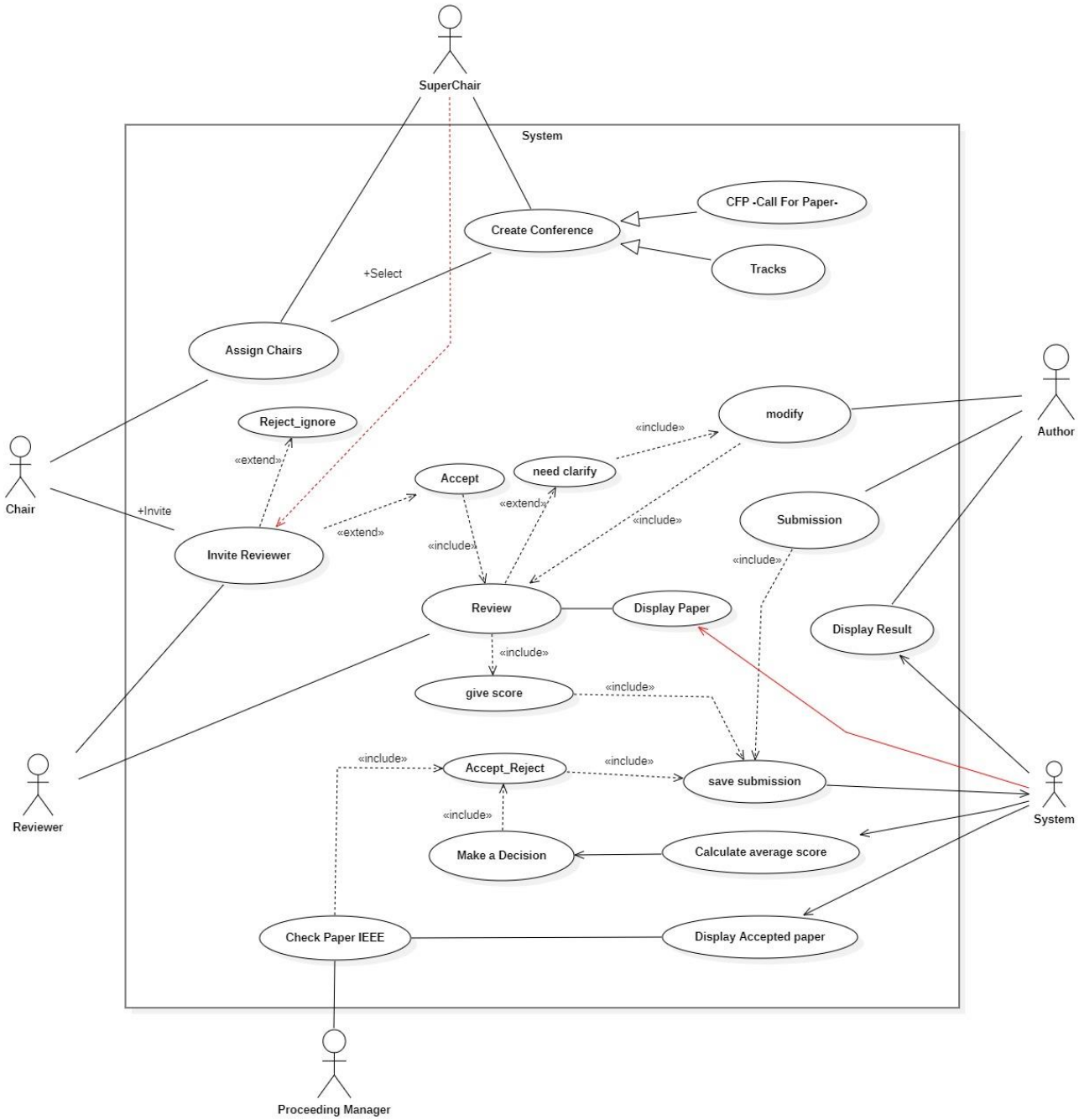


*Figure 3 Use Case Diagram*

### 3.1.2. Use case Specification:

A Use Case Specification is **a textual description of the functionality provided by the system**. It captures actor-system interaction. That is, it specifies how a user interacts with a system and how the system responds to the user actions. In this section I will explain the use case according to tables to make it easier to understand.

Tables have a 4 main title:

- Name: name of use case which I will explain in the table.
- Actor: name of actor that responsible for this use case.
- Description: some information about use case.
- Main course: the purpose and the expecting target from this use case.

### 3.1.3. Use Case Tables:

**Table 1 Create Conference:**

| | |
|---|---|
| **Name** | Create conference |
| **Actor** | Super Chair |
| **Description** | The step where the user wants to hold a conference and start preparing for it. |
| **Main Course** | Step 1: choose a name and title to conference that's indicates to the main subject of conference.<br>Step 2: start track insertion (Ex: if you have a conference about information technology you should have a tracks such as: AI, CS, Programming language …*etc*.)<br>Step 3: wait until admin approved your request. |

**Table 2 Assign Chair:**

| | |
|---|---|
| **Name** | Assign Chair |
| **Actor** | Super Chair |
| **Description** | The step where super chair start selecting the users responsible for select reviewer. |
| **Main Course** | Find the best user who specializes in the selected path and choose to manage it. |

**Table 3 Invite Reviewer:**

| | |
|---|---|
| **Name** | Assign Chair |
| **Actor** | Chair |
| **Description** | notice of new changes. |
| **Main Course** | Step 1: Super chair must find someone specialized with specific path “Track” and give him a new role that meaning he have a new access control.<br>Step 2: notice of new changes must be sent to the Chair, describing his responsibility. |

**Table 4 Invite Reviewer:**

| | |
|---|---|
| **Name** | Invite Reviewer |
| **Actor** | Chair, Reviewer |
| **Description** | Find specialized and invite there to be a reviewer |
| **Main Course** | Step 1: chair must find someone specialized with specific path “Track” and Invite him by (E-mail, SMS, Invitation Link, Social media, … *etc*) to be a reviewer.<br>Step 2: Reviewer can accept, Reject or ignore invitation.<br>Step 3: If Reviewer accept invitation |

**Table 5 Review:**

| | |
|---|---|
| **Name** | Review |
| **Actor** | Reviewer |
| **Description** | Review paper to determine if paper acceptable into conference or not. |
| **Main Course** | Step 1: reviewer go through access control to see new paper review request (The paper shown must be in the track in which he is majoring).<br>Step 2: give score and feedback.<br>Step (optional): Reviewer can give author hint or modification note.<br>Step 3: submit.<br>Step 4: submission goes through calculation process in system to calculate average paper score. |

**Table 6 Submission:**

| | |
|---|---|
| **Name** | Submission |
| **Actor** | Author |
| **Description** | Use case to give user authority to submit the paper. |
| **Main Course** | Step 1: Select track he wants to participate with.<br>Step 2: Upload paper with file extension (PDF, DOCX, PPTX … *etc*.)<br>Step 3: write description.<br>Step 4: submit paper. |

**Table 7 Save submission:**

| Name | Save submission |
|---|---|
| Actor | System, Author, Reviewer, Chair |
| Description | Use case send submission in website to system database. |
| Main Course | Step 1: Save submission into database.<br>Step 2: Display necessary data to the right users (By user role). |

**Table 8 Check Paper:**

| Name | Check Paper IEEE |
|---|---|
| Actor | Proceeding manager |
| Description | Proceeding manager must make a decision to determine if paper acceptable into conference or not. |
| Main Course | Step 1: Proceeding manager go through access control to see papers (The paper shown must be in the track in which he is majoring).<br>Step 2: choose paper to check.<br>Step 3: check if there any copyright or AI generator.<br>Step 4: submit his checking result. |

### 3.1.4. Actors Tables:

**Table 1 SuperChair:**

| | |
|---|---|
| **Actor** | SuperChair |
| **Description** | Actor responsible for Create conference -which having (Tracks, Date, Name, Title)- and assign chair for each track. |

**Table 2 Chair:**

| | |
|---|---|
| **Actor** | Chair |
| **Description** | Actor responsible for Invite reviewer for his majoring and see paper review to decide if it acceptable or not. |

**Table 3 Reviewer:**

| | |
|---|---|
| **Actor** | Reviewer |
| **Description** | Actor responsible for seeing the paper and giving it the score and feedback to present to the Chair. |

**Table 4 Author:**

| | |
|---|---|
| **Actor** | Author |
| **Description** | A simple user wants to participate in the conference which it done by submitting research paper. |

**Table 5 System:**

| | |
|---|---|
| **Actor** | System |
| **Description** | System is a server having a database to save submission and sharing data or resources among multiple clients or performing. computations for a client. [] |

**Table 5 System:**

| | |
|---|---|
| **Actor** | Proceeding Manager |
| **Description** | Actor Responsible for check copyrights and IEEE standards |

## 3.2. Class:

### 3.2.1. Class diagram:

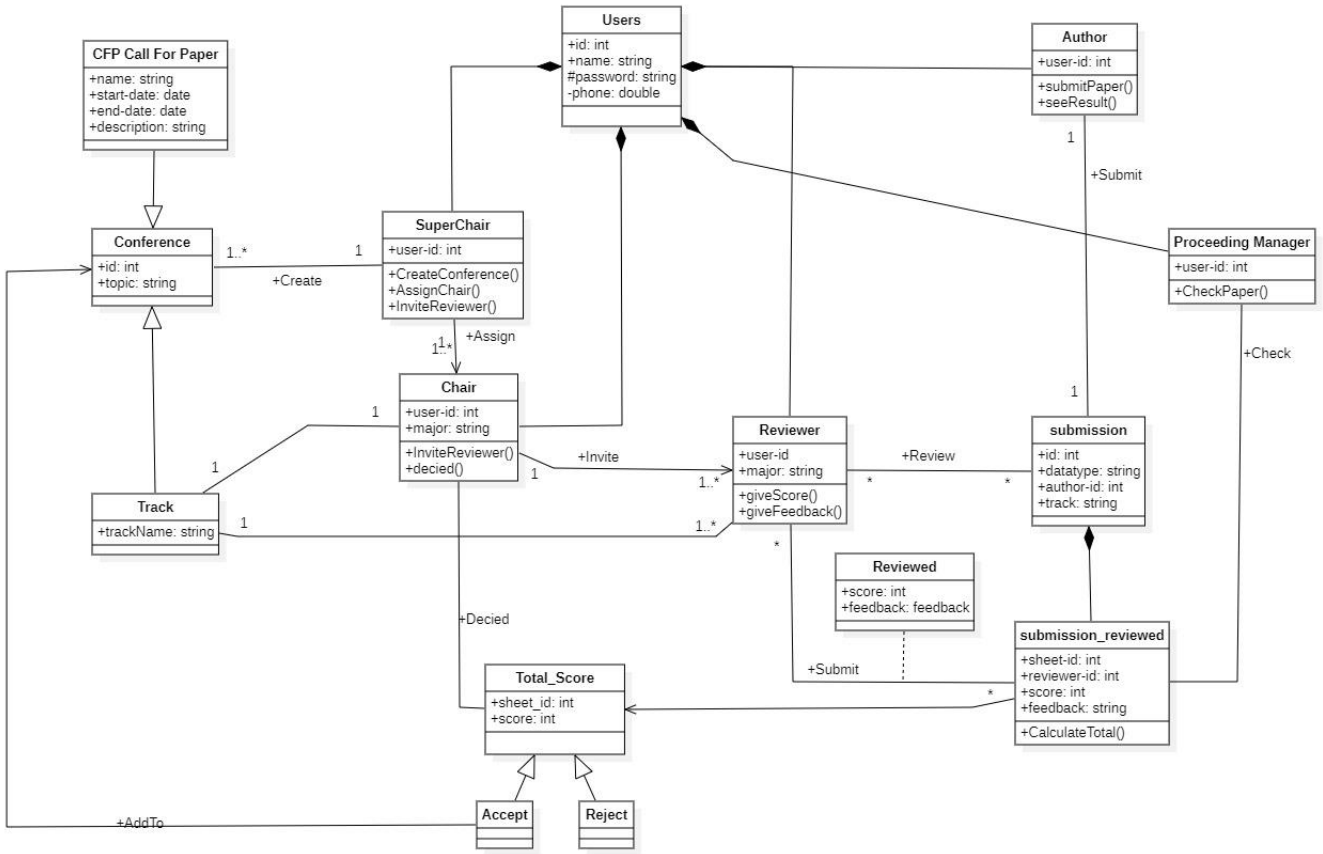


*Figure 4 Class Diagram*

### 3.2.2. Class Diagram Specification:

**Meaning:**

In software engineering, a class diagram in the Unified Modelling Language (UML) is **a type of static structure diagram** that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. []

**Purpose:**

- Shows static structure of classifiers in a system.
- Diagram provides a basic notation for other structure diagrams prescribed by UML.
- Helpful for developers and other team members too
- Business Analysts can use class diagrams to model systems from a business perspective. []

## 3.3. Activity:

### 3.3.1. Activity Diagram:

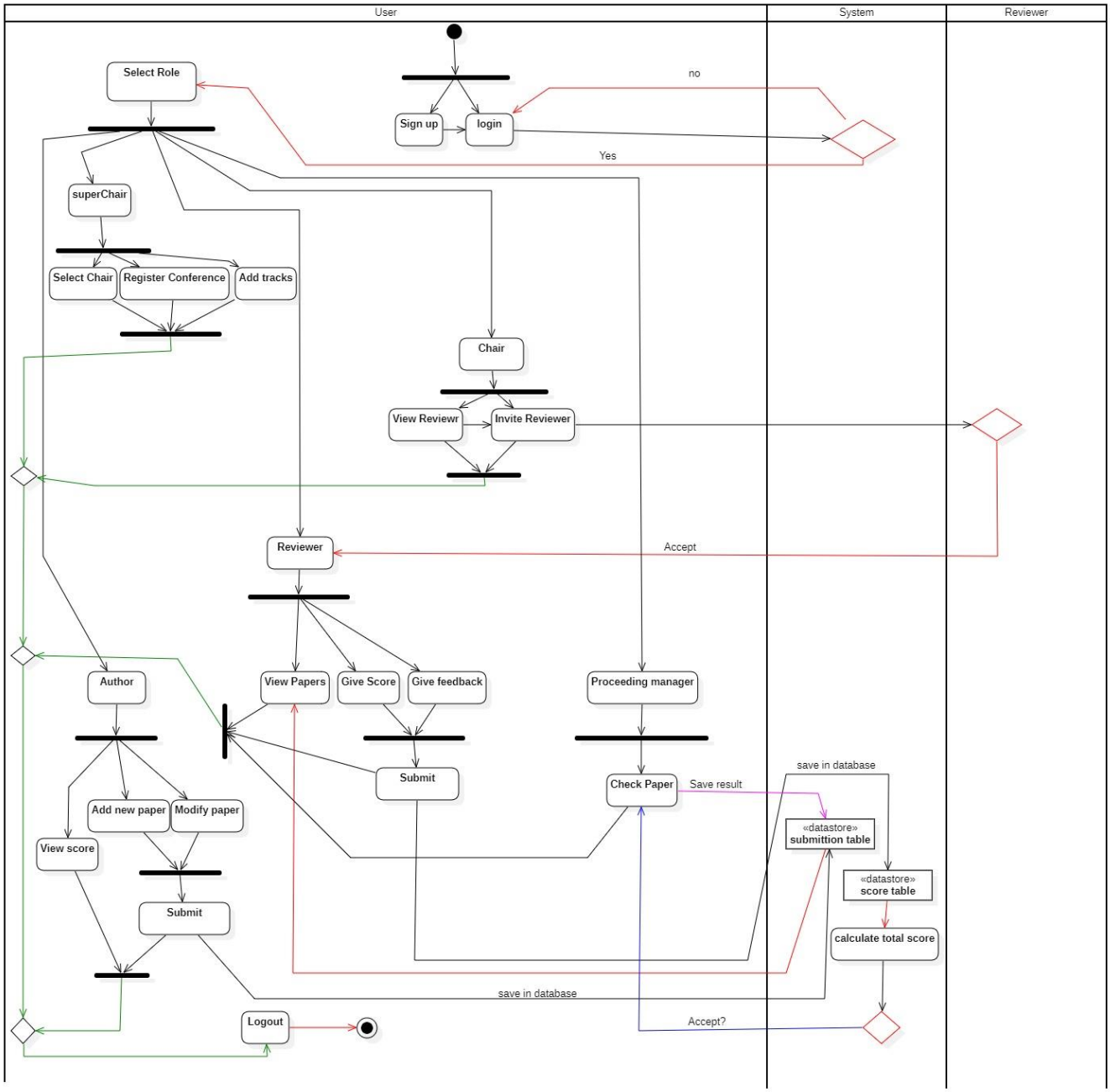


*Figure 5 Activity Diagram*

### 3.3.2. Activity Diagram Specification:

**Meaning:**

Activity diagram is another important behavioural diagram in UML diagram to describe dynamic aspects of the system. Activity diagram is essentially an advanced version of flow chart that modelling the flow from one activity to another activity. []

**Purpose:**

- **Modelling Workflow:** Activity diagrams are commonly used to model and represent the workflow of a system or a business process. They provide a visual representation of the sequence of activities and how they are interconnected.
- **Process Modelling:** They are effective for modelling complex processes, showing the activities involved and the order in which they occur. This helps in understanding the logic and dependencies between different tasks.
- **Behavioural Modelling:** Activity diagrams help in describing the dynamic aspects of a system by illustrating how different activities are performed and how they interact with each other.
- **Communication:** Activity diagrams serve as a communication tool between stakeholders such as analysts, developers, and users. They provide a clear and understandable visual representation of the system's behaviour.
- **Analysis and Design:** During the system analysis and design phase, activity diagrams assist in understanding the functional aspects of a system and in making design decisions related to the flow of control.
- **Documentation:** Activity diagrams can be used as part of system documentation to document the business processes or operational procedures within an organization.
- **Identifying Parallelism and Synchronization:** They are useful for identifying parallel activities and synchronization points in a process. This helps in analysing concurrency and potential bottlenecks.
- **Verification and Validation:** Activity diagrams can be employed for verifying and validating the correctness of a system's behaviour by representing different scenarios and paths through the activities.

## 3.4. Sequence:

### 3.4.1. Sequence Diagram:

**SuperChair:**

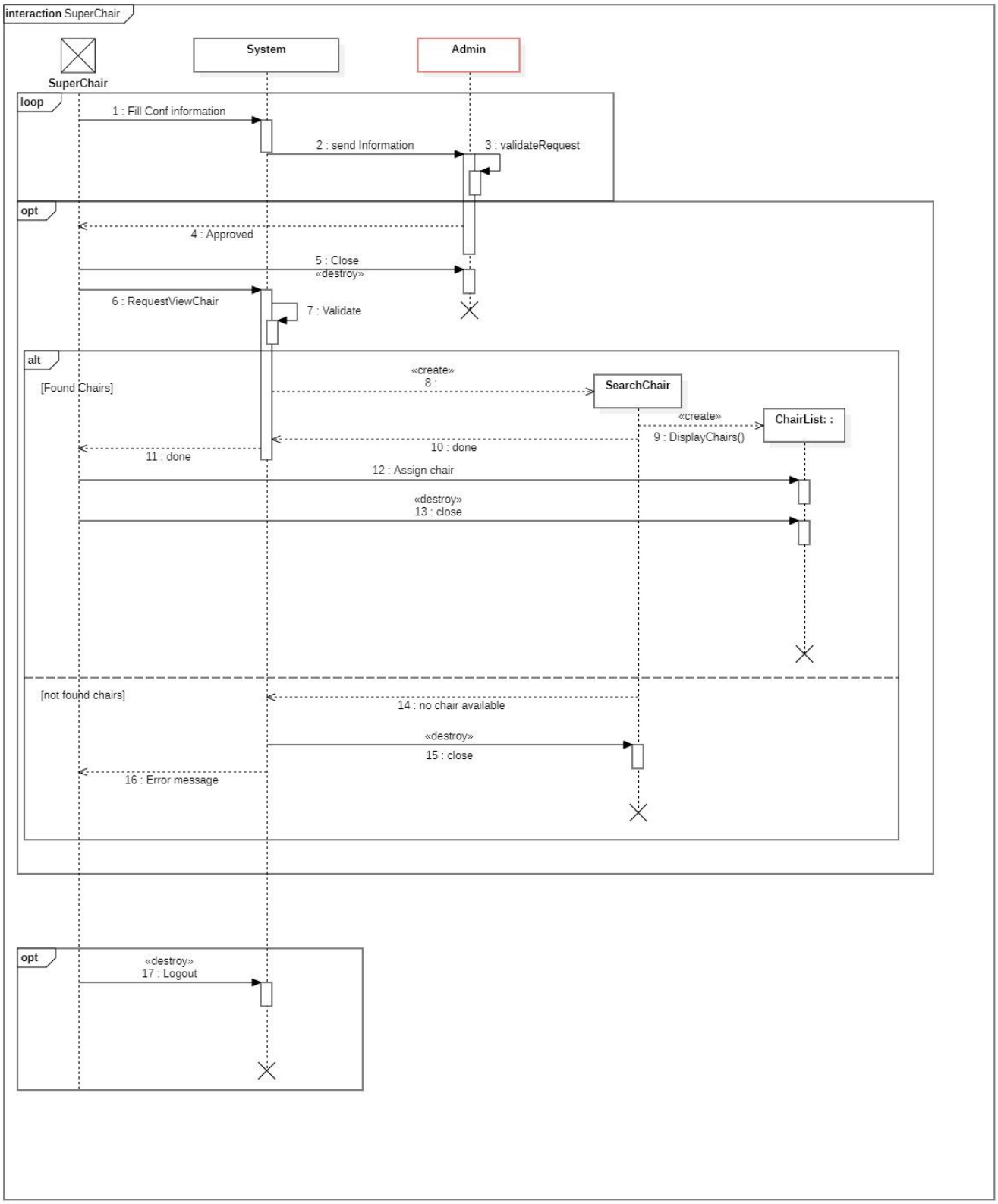


*Figure 6 SuperChair Sequence Diagram*

## Chair:

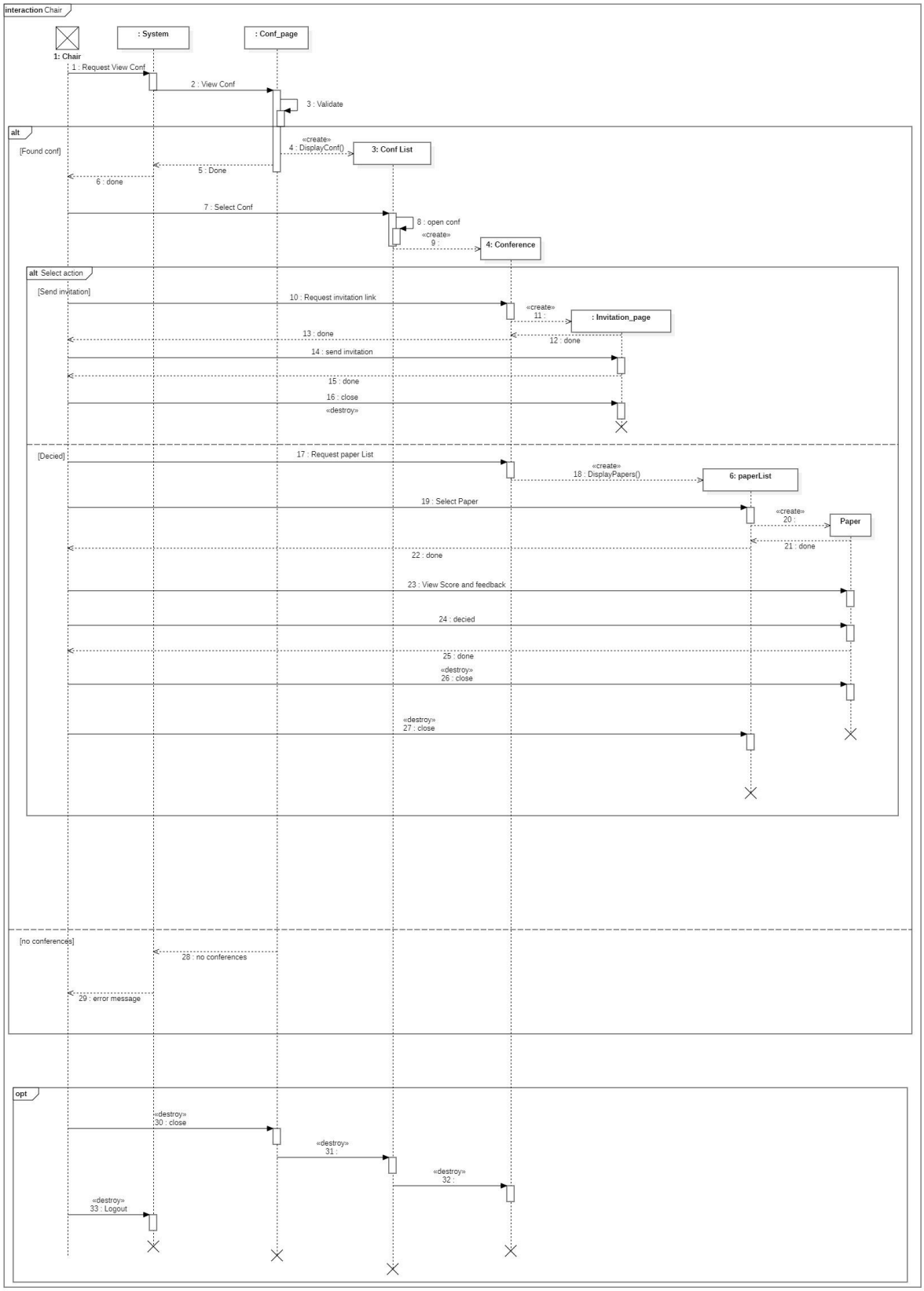


*Figure 7 Chair sequence Diagram*

## Reviewer:

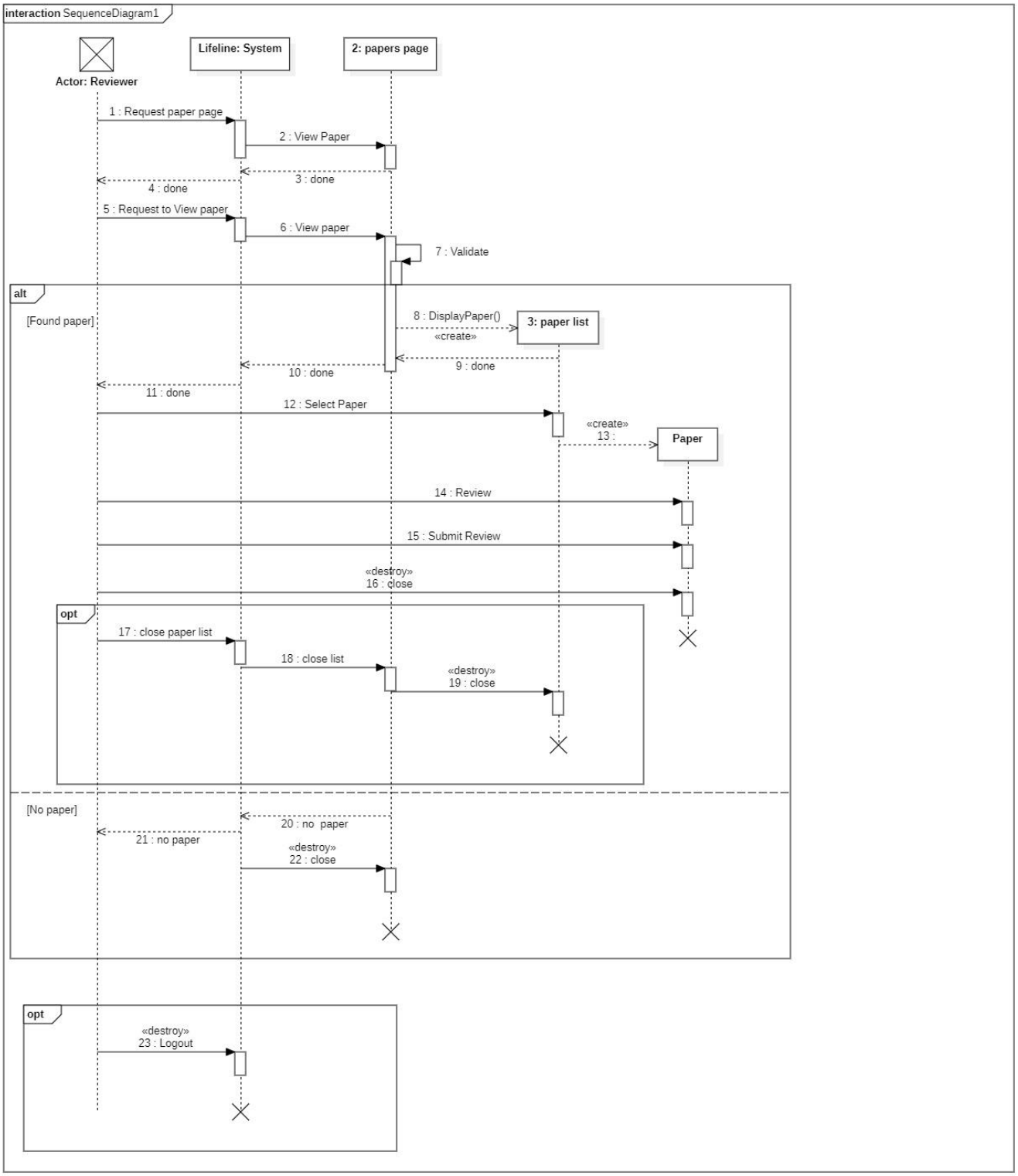


*Figure 8 Reviewer Sequence Diagram*

## Author:

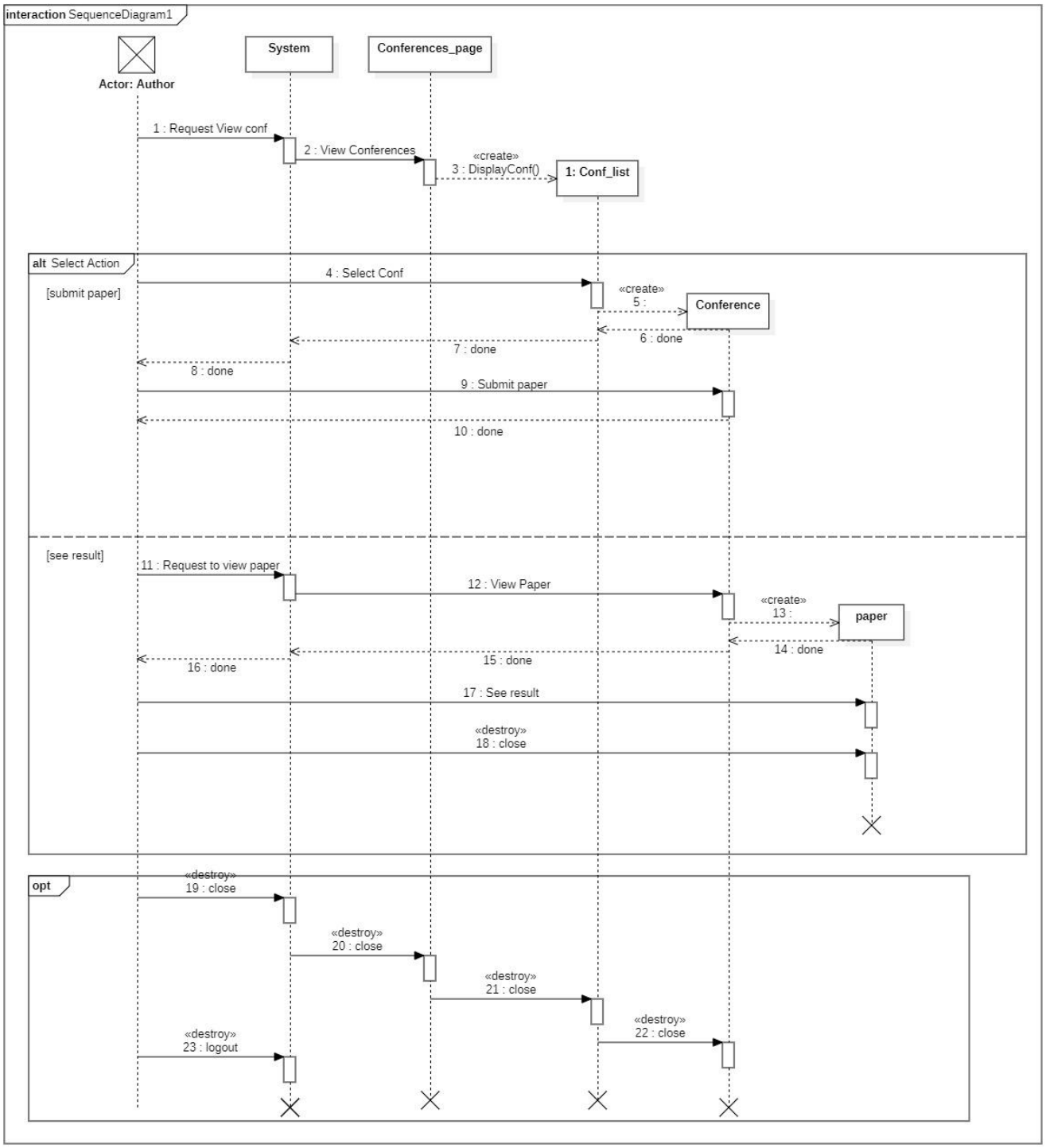


*Figure 9 Author Sequence Diagram*

## Proceeding Manager:

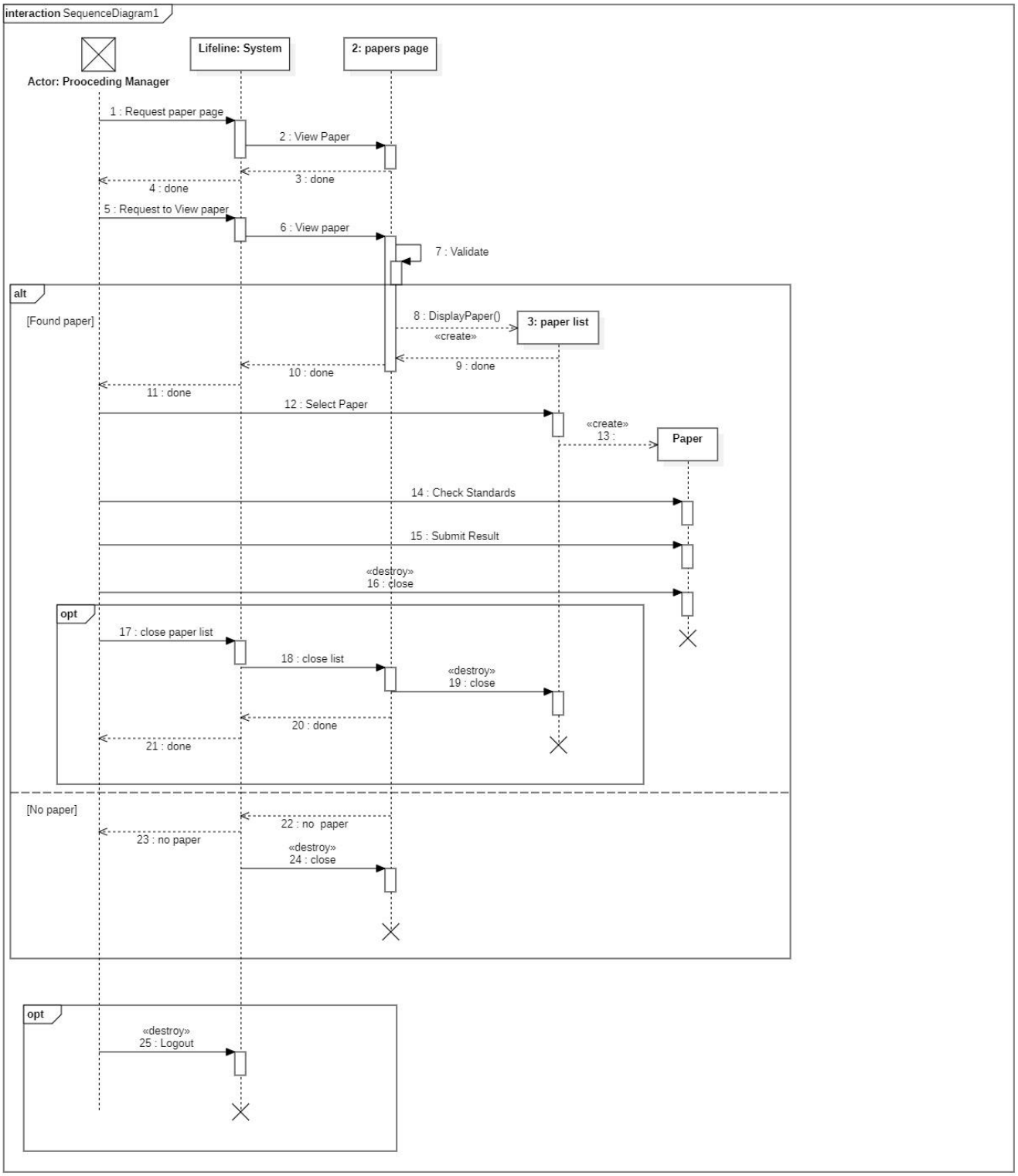


*Figure 10 Proceeding Manager Sequence Diagram*

### 3.4.2. Sequence Diagram Specification:

**Meaning:**

A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. []

**Purpose:**

- **Visualizing Interactions:** Sequence diagrams offer a visual representation of the interactions and communication between different objects or components within a system. They show the order in which messages are exchanged, providing a clear understanding of the flow of control.
- **Modelling System Behaviour:** Sequence diagrams help in modelling and capturing the dynamic behaviour of a system by showing how objects collaborate and communicate with each other during a particular scenario or use case.
- **Understanding Message Flow:** They depict the flow of messages between objects, including the timing and order of these messages. This is particularly useful for understanding the sequence of actions and responses in a system.
- **Identifying Collaboration:** Sequence diagrams identify the collaboration and cooperation between objects, making it easier to analyse the relationships and dependencies in a system.
- **Analysing Scenarios:** Sequence diagrams are valuable for analysing specific scenarios or use cases by illustrating the interactions between objects in a step-by-step manner. This aids in understanding the behaviour of the system under different conditions.
- **Communication and Documentation:** They serve as a communication tool between stakeholders, including analysts, designers, and developers. Sequence diagrams are used for documenting and conveying the details of system interactions and message flows.
- **Design and Implementation Guidance:** Sequence diagrams can assist in the design and implementation phases by providing insights into the interactions and dependencies among different components. They help developers understand how to implement the desired behaviour.
- **Identifying Potential Issues:** By visualizing the interactions, sequence diagrams can help identify potential issues, such as message delays, missing messages, or unexpected behaviour in the system.
- **Testing and Validation:** Sequence diagrams can be used as a basis for testing and validating the correctness of a system's interactions. They provide a reference for expected behaviour during testing.

# Design phase

## 1. Entity Relationship:

### 1.1. Entity Relationship Diagram:

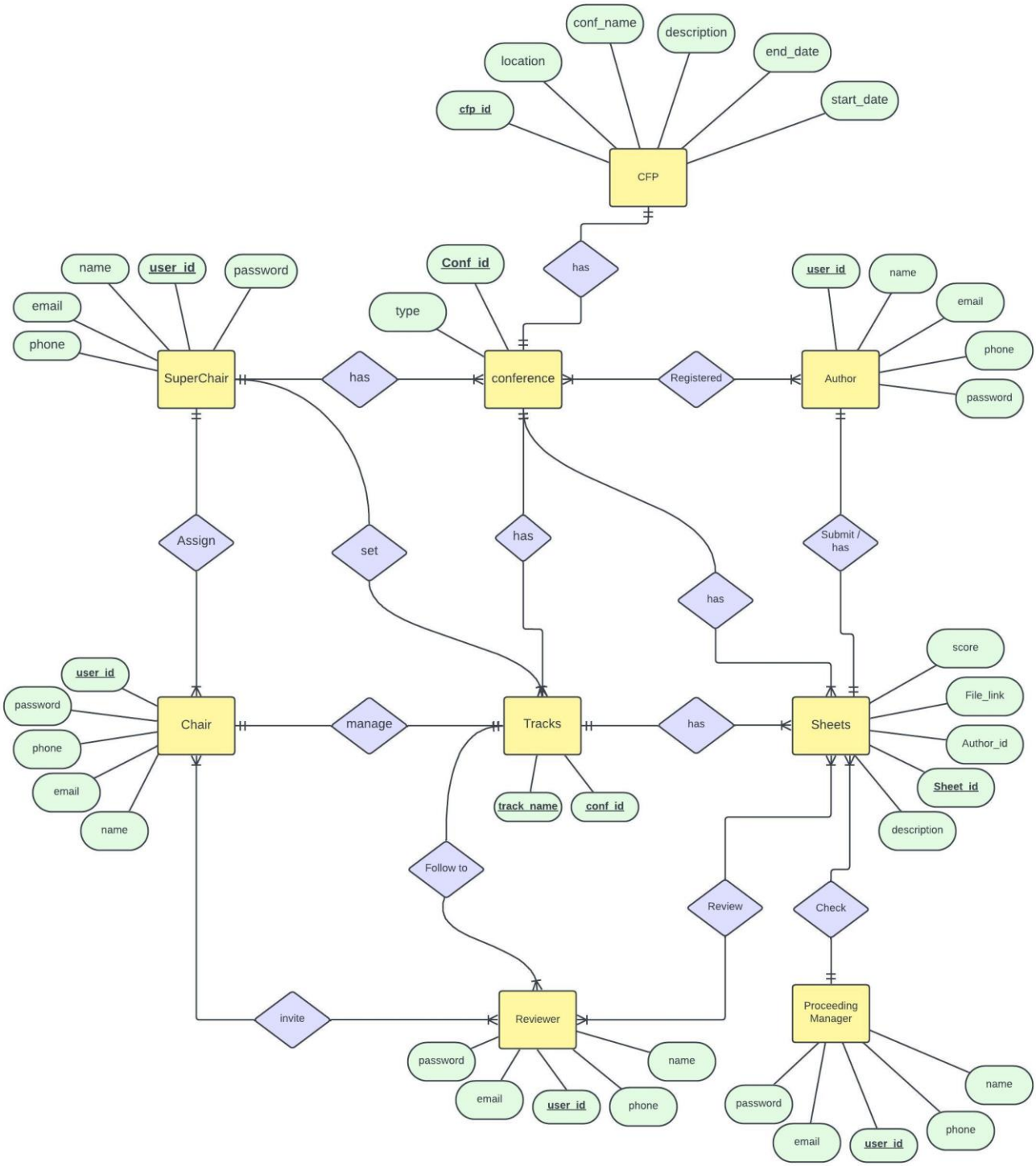


*Figure 11 Entity Relationship Diagram*

## 1.2. Entity Relationship Specification:

### 1.2.1. Meaning:

entity relationship model, is a graphical representation that depicts relationships among people, objects, places, concepts or events within an information technology (IT) system. An ERD uses data modelling techniques that can help define business processes and serve as the foundation for a relational database.[]

### 1.2.2. Necessary information:

I have ignored two main entities (User entity, Role entity) to make the diagram easier to read.

This System based on roles, So user entity must have all information about the user such as:

Name, password, email, phone number, …*etc.*

And role entity must have users with their role.

We have ignored these entities because of the relationships that require showing the flow of the system because the diagram would be difficult to read and in the two cases would give us the same result.

We have ignored the entities in the entity diagram only not from the entire system, they will be in the database schema.

# 2. Database Design:

## 2.1. Database schema:

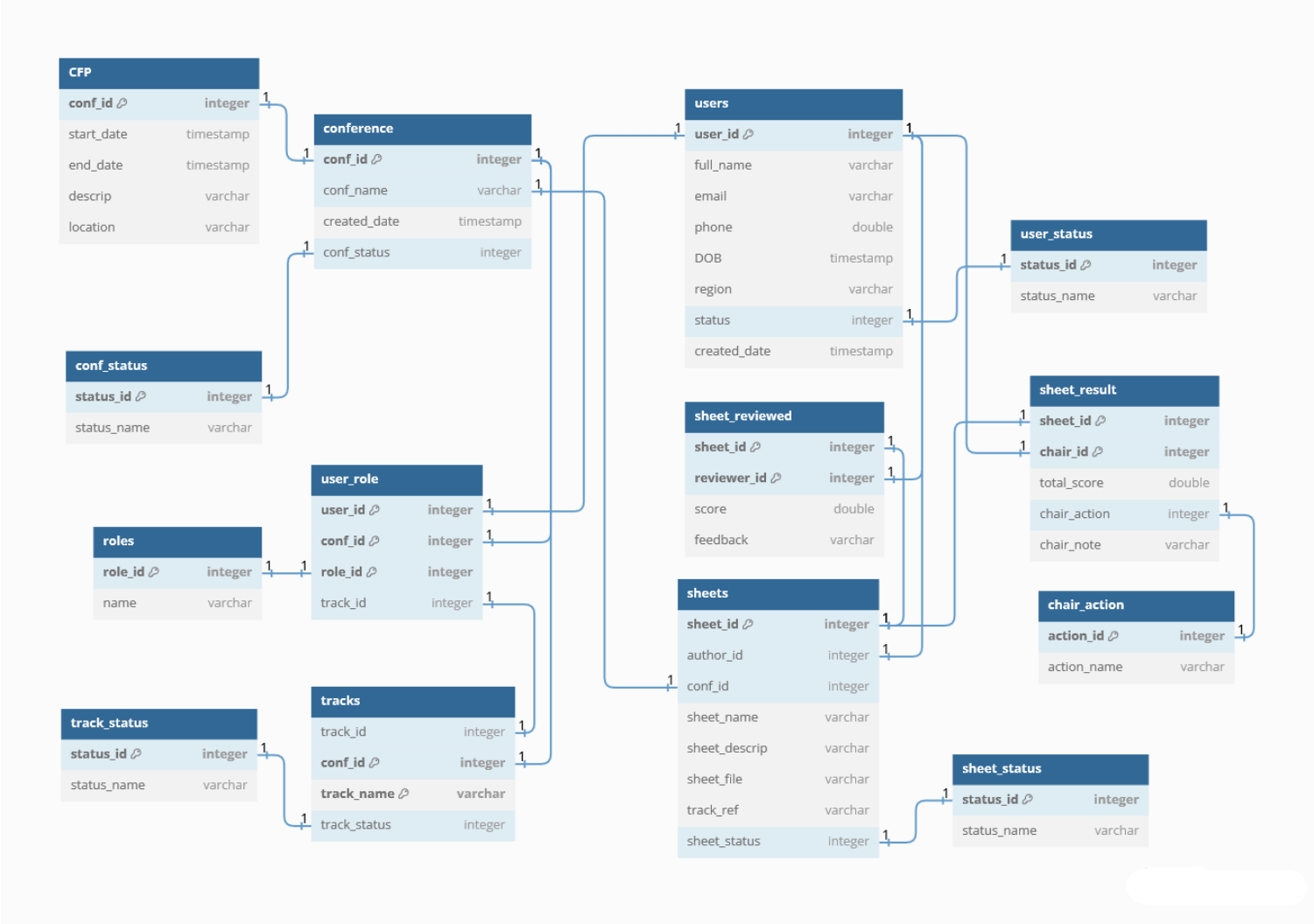


*Figure 12 Database Schema*

## 2.2. Database Schema specification:

### 2.2.1. Meaning:

A database schema defines how data is organized within a relational database; this is inclusive of logical constraints such as, table names, fields, data types, and the relationships between these entities. Schemas commonly use visual representations to communicate the architecture of the database, becoming the foundation for an organization's data management discipline. []

### 2.2.2. Status in schema:

Status tables in schema assist database administration to monitoring database process, and conference administration to manage users and sheets.

What's status tables?
the state which object is, So we can take a user as an example:
status table should have 3 status:

- Active.
- Inactive.
- Blocked.

Each status has an ID, So in users table we can use status id to identify user state.

# GUI

## 1. Screenshots:

### 1.1. Home page:

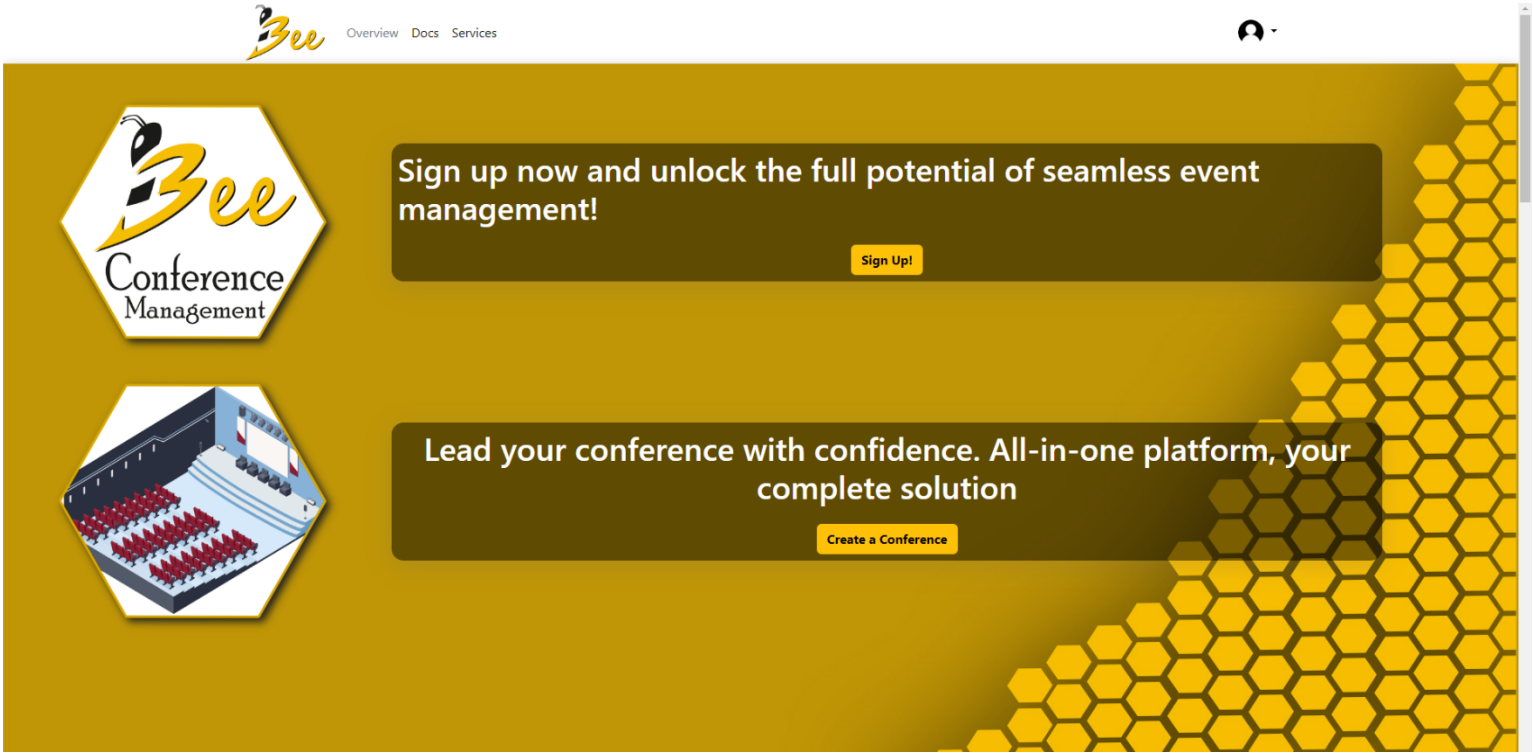


*Figure 13 home screen 1*

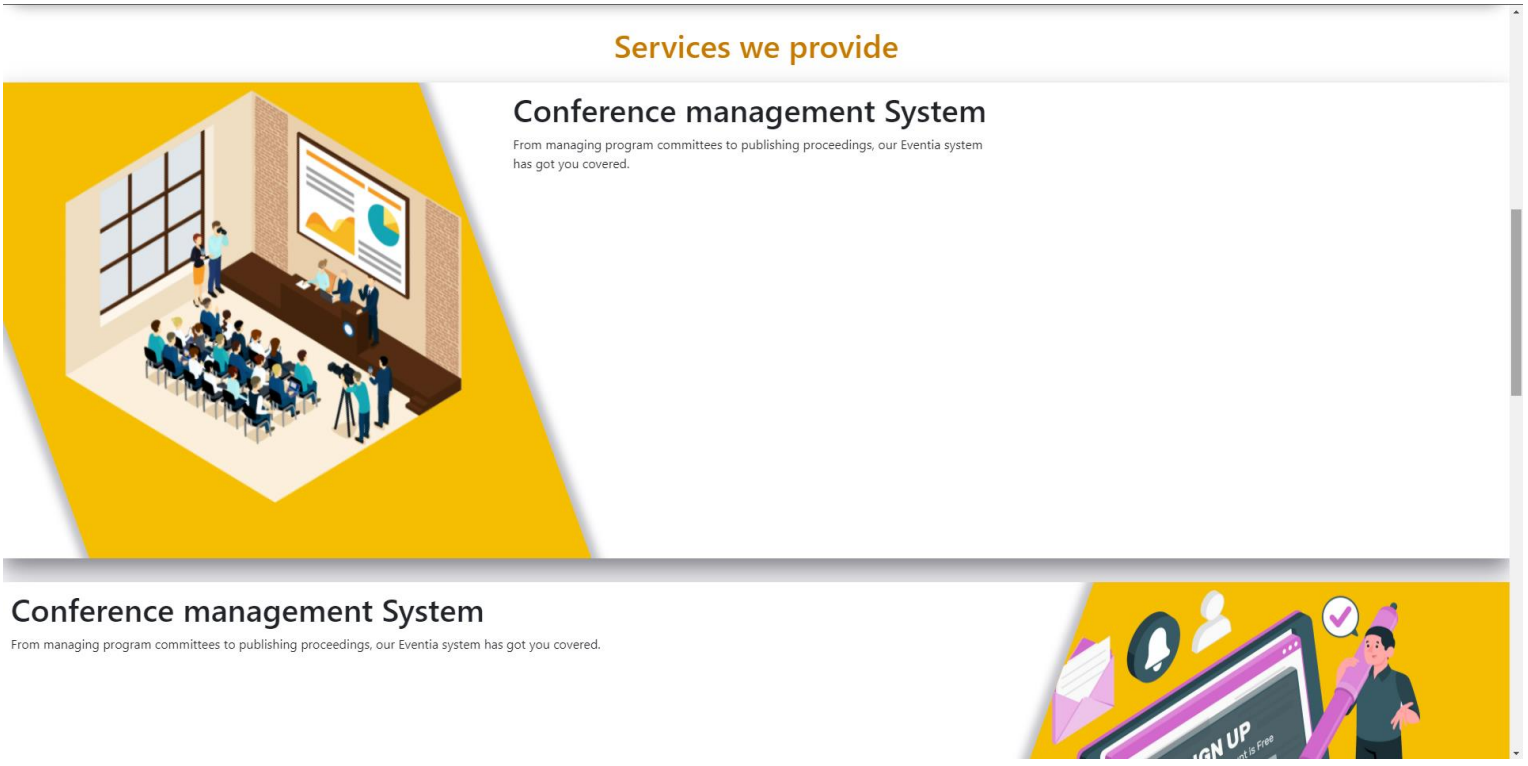


*Figure 14 home screen 2*

## 1.2. Find CFPs page:

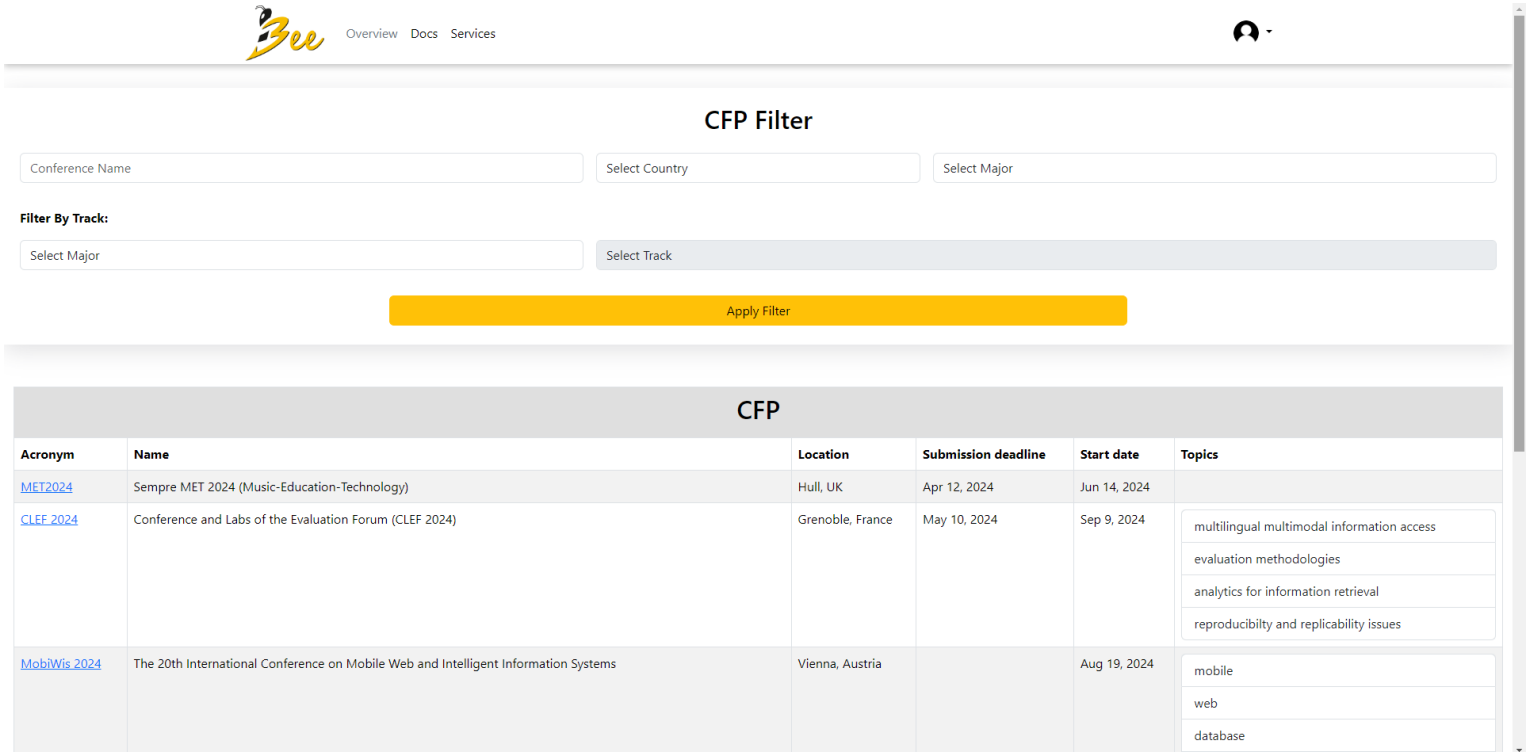


*Figure 15 CFP page*

## 1.3. Conference CFP:

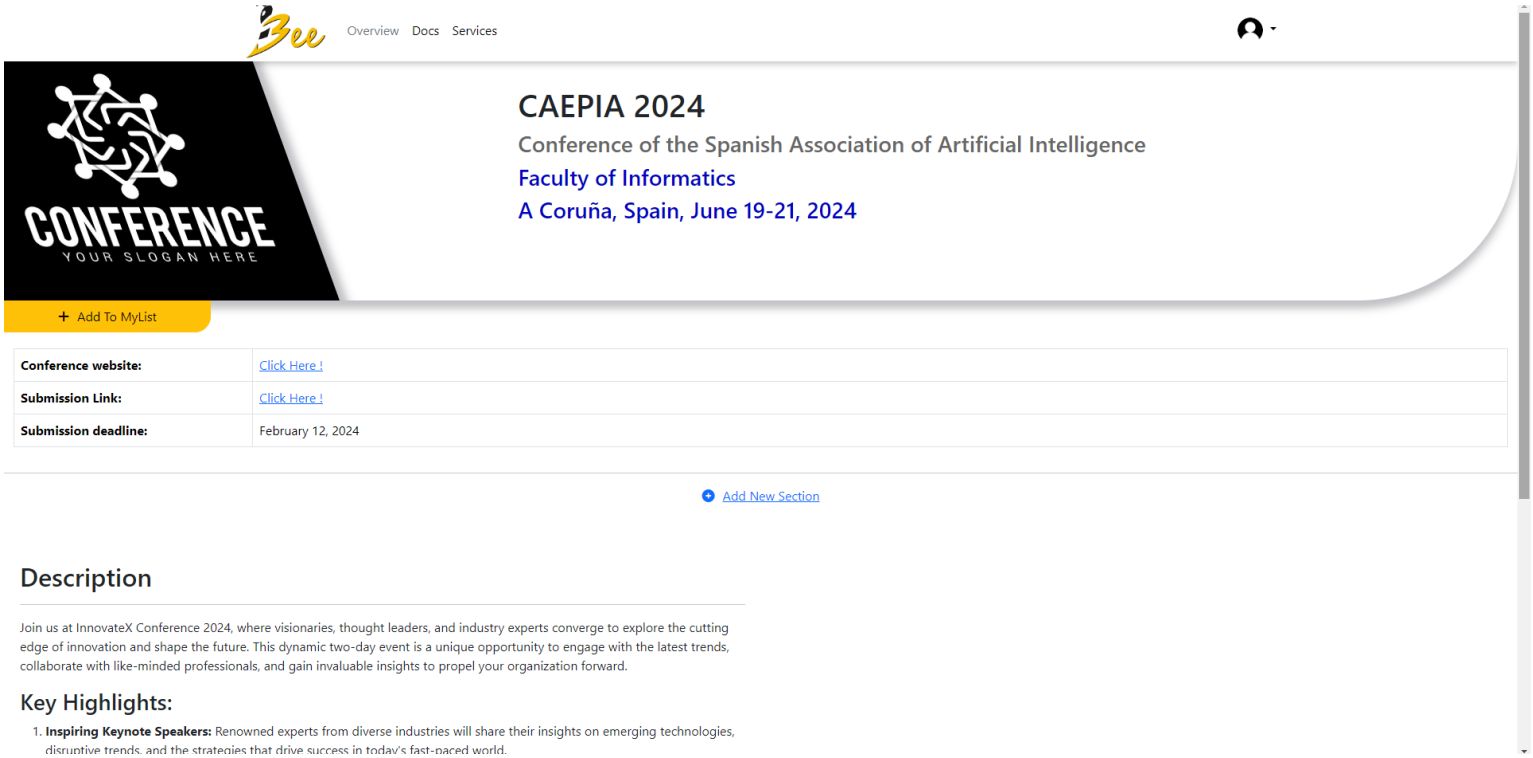


*Figure 16 conference CFP page*

## 1.4. Create CFP page:

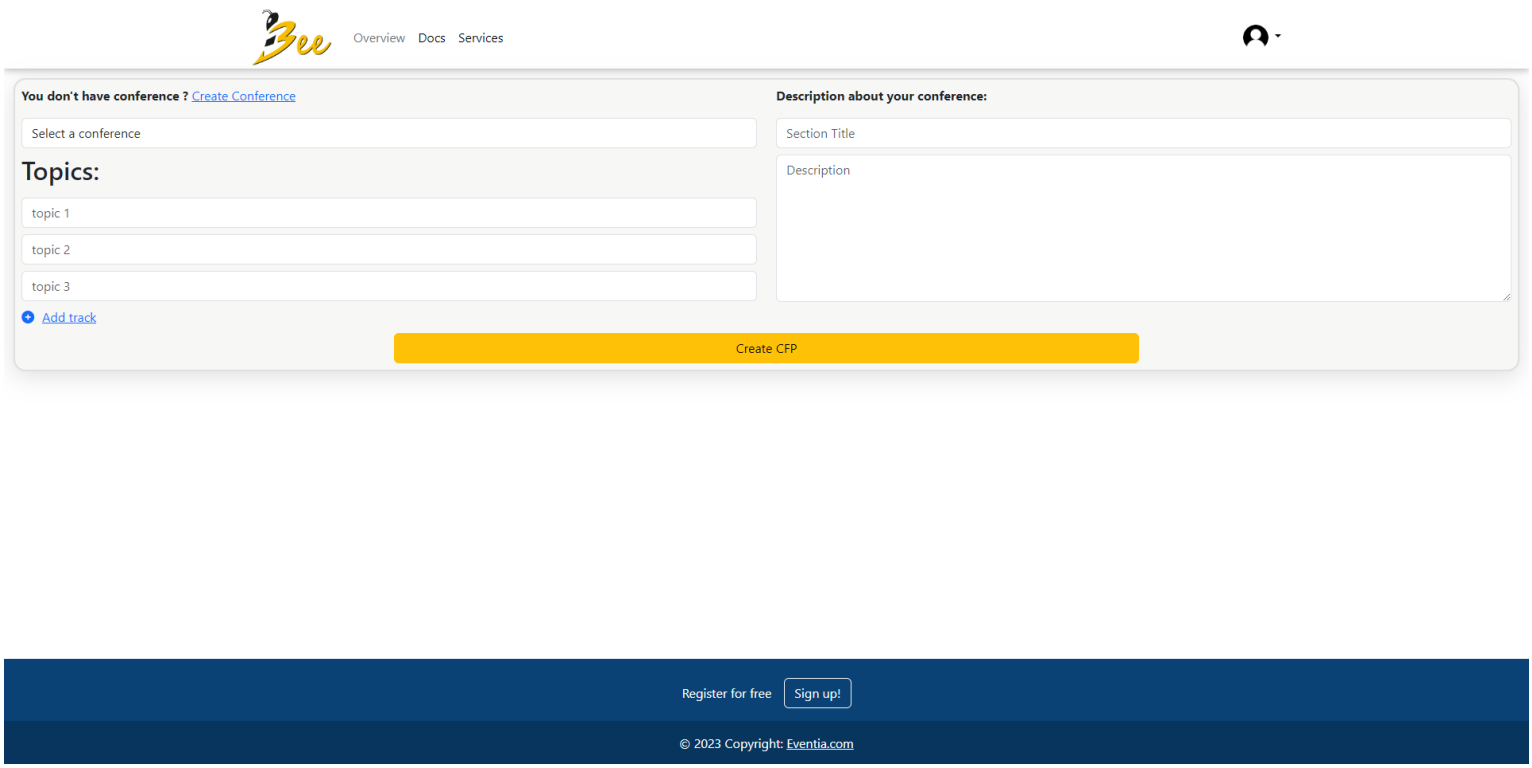


*Figure 17 Create CFP page.*

## 1.5. Create Conference page:

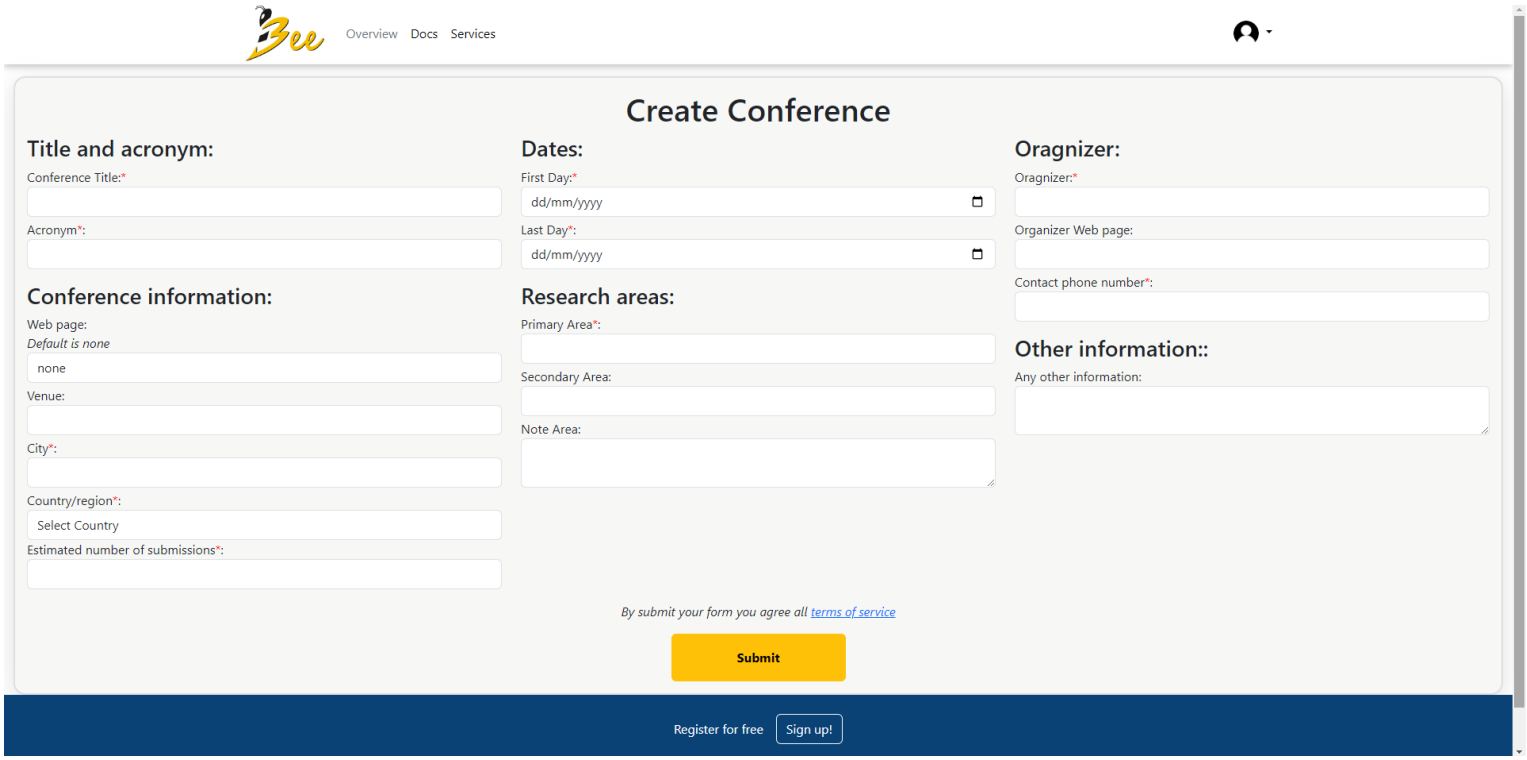


*Figure 18 Create Conference Page*

## 1.6. Select Role page:

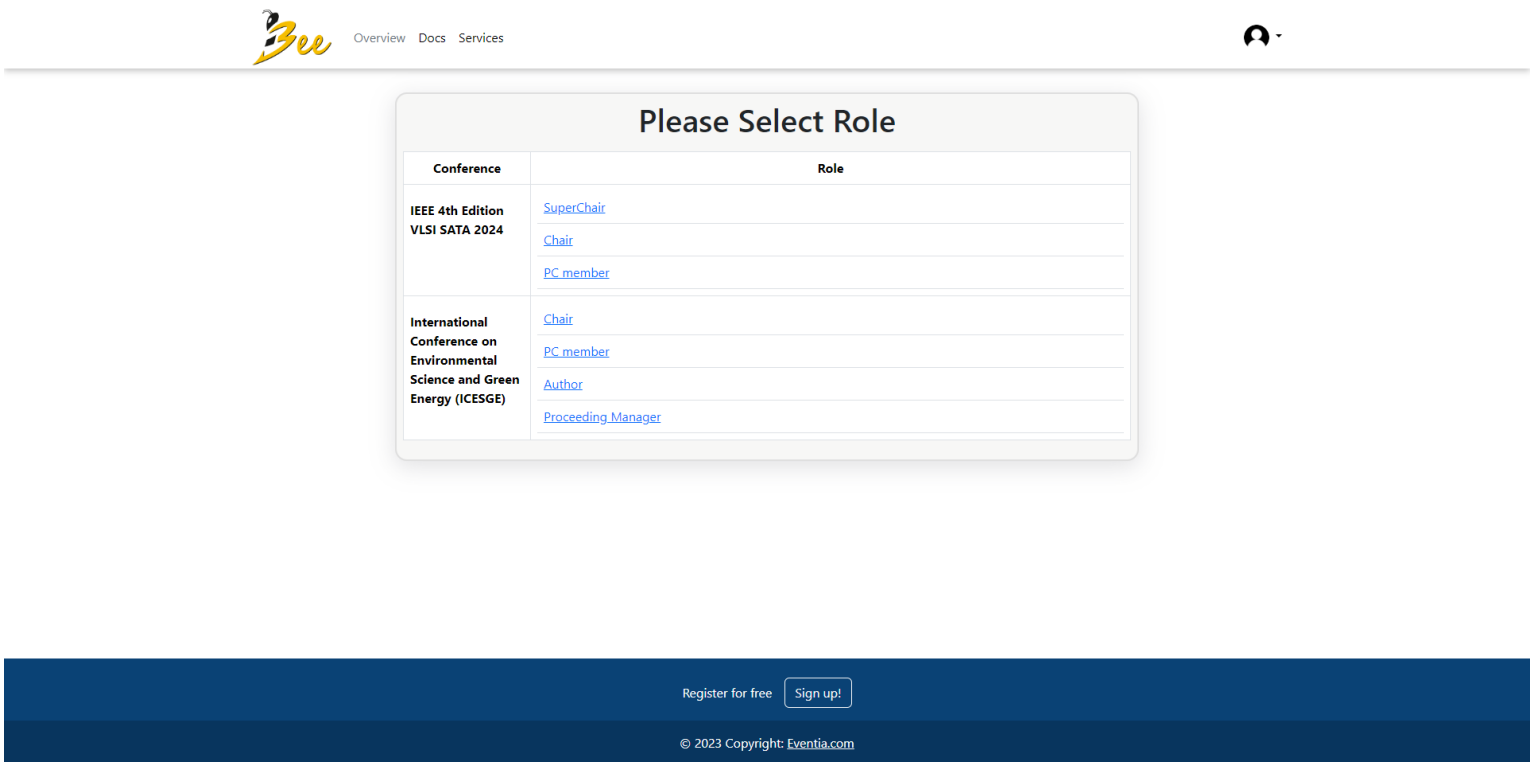


*Figure 19 Select Role page.*

## 1.7. Submit Paper page:

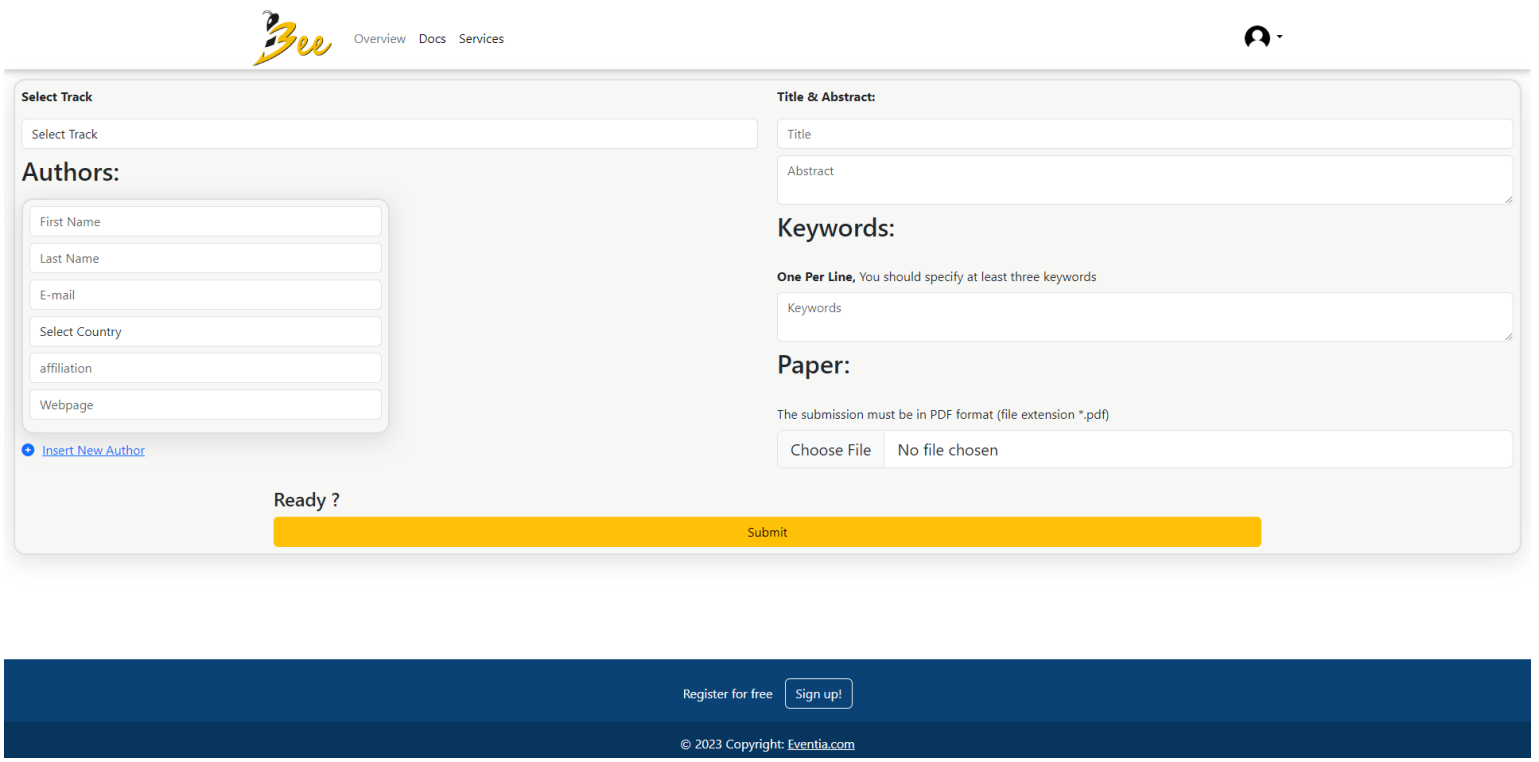


*Figure 20 Submit Paper page.*

## 1.8. Account page:

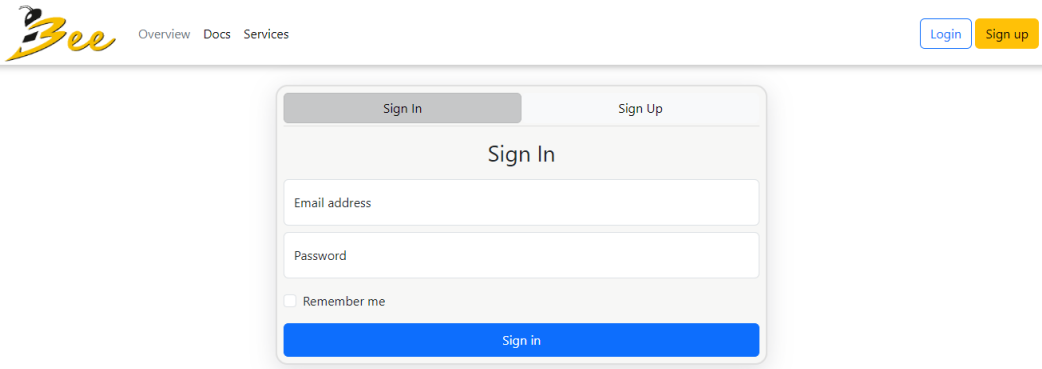


*Figure 21 Sign In page*

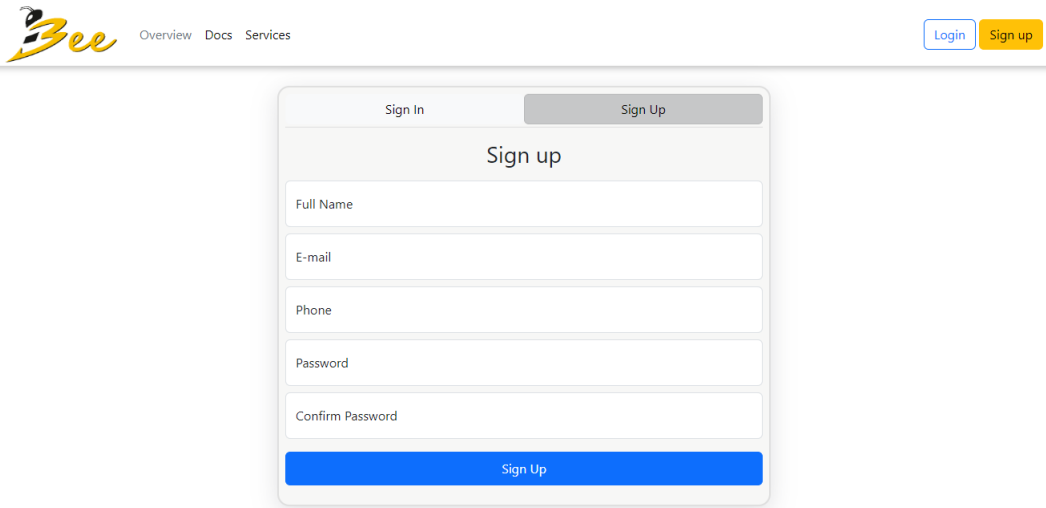


*Figure 22 Sign Up page*

# References


1. **Links:**

- https://en.wikipedia.org/wiki/Server_(computing)
- https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/
- https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/
- https://www.lucidchart.com/pages/uml-sequence-diagram
- https://www.techtarget.com/searchdatamanagement/definition/entity-relationship-diagram-ERD
- https://www.lucidchart.com/
- https://dbdiagram.io
- https://www.ibm.com/topics/database-schema
- https://easychair.org/
- https://easychair.org/overview
- https://en.wikipedia.org/wiki/EasyChair
- https://www.openconf.com/
- https://en.wikipedia.org/wiki/OpenConf

# Appendices

## 1. Appendix A Code:

### 1.1. Attachments Code:

HTML:

- home/index.html
- account/index.html
- cfp/index.html
- conf_cfp/index.html
- create_cfp/index.html
- select_role/index.html
- submit_paper/index.html

CSS:

- CSS/style.css

JavaScript/jQuery:

- JS/account.js

(BOOTSTRAP 5 Library) JavaScript, CSS

(AOS Library) JavaScript, CSS

## 2. Appendix B Schedule:

### 2.1. Table:

| # | Date | Description |
|---|---|---|
| 1 | January 4 | Introductory meeting with Eng. Mustafa |
| 2 | January 11 | Discuss our work on chapter 1,2,3 and how to system works. |
| 3 | January 23 | Revision of overall documentation |